FireWire in Depth

Session 115

FireWire in Depth

Eric Anderson
Manager, FireWire Software

Introduction

- This session is about writing drivers and applications for FireWire devices
    - Basic architecture and service collection
    - What has changed since 2001?
    - Where is FireWire going in 2002?
- New “How-To” section for various device types

What You Will Learn

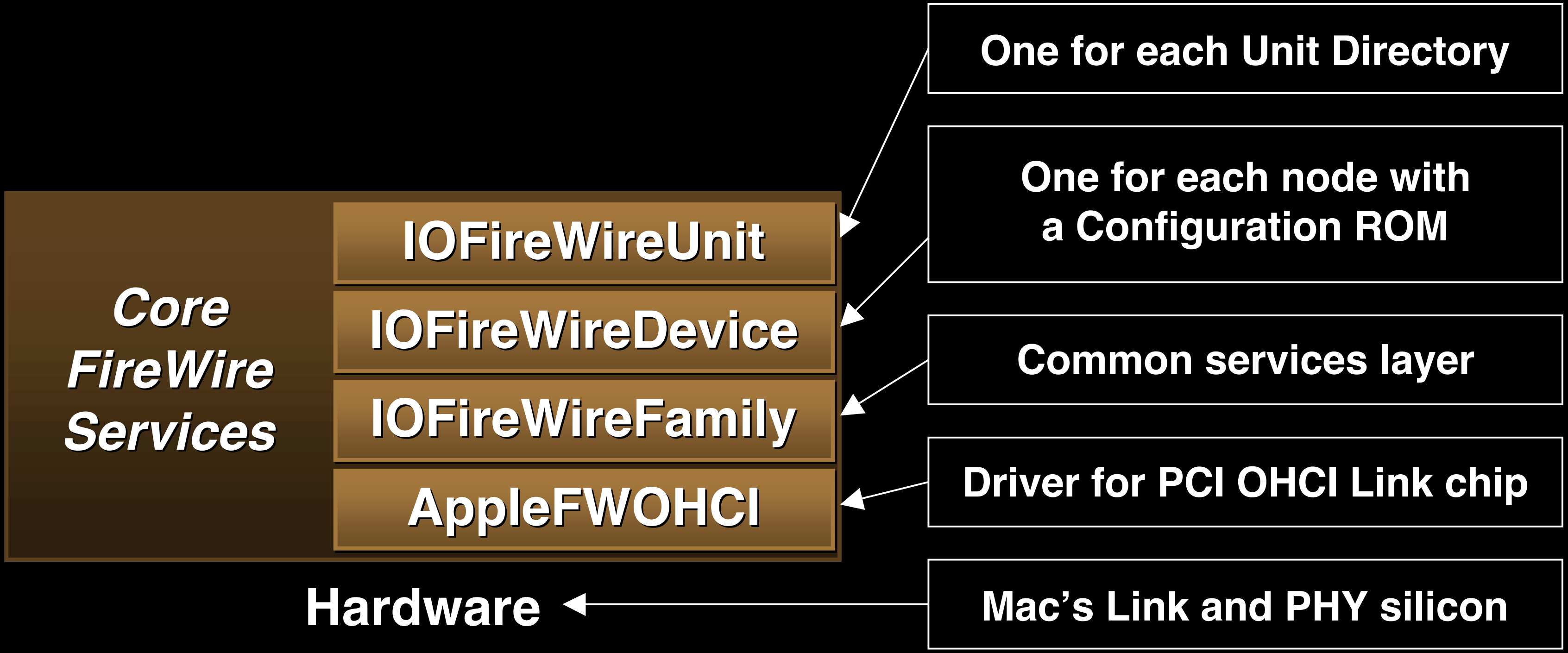
- FireWire services in Mac OS X
- Changes since WWDC 2001
- Planned future services
- Device and driver How-Tos
- Developer resources

FireWire Services in Mac OS X

FireWire in the Kernel

- FireWire is a kernel service
  - This permits booting from FireWire drives
  - Exists within the IO Kit architecture
  - Implemented as Kernel Extensions (KEXTs)
- Most FireWire drivers belong in application layer ("user space")—not in the kernel
  - Developing outside the kernel is much easier

Core FireWire Services



Core FireWire Services
IOFireWireUnit
IOFireWireDevice
IOFireWireFamily
AppleFWOHCI
Hardware
One for each Unit Directory
One for each node with a Configuration ROM
Common services layer
Driver for PCI OHCI Link chip
Mac’s Link and PHY silicon

IOFireWireFamily

- Core service layer (IOFireWireFamily.kext)
- Multiplexes devices and drivers that need to cooperatively share a single FireWire bus
  - Example: DV camera and FW disk drive
- Manages common FireWire services
  - Bus scan and driver matching
  - GUID and topology tracking
  - Packet transmit and receive

Driver Matching and Loading

- FireWire populates the IORegistry
- IO Kit does actual matching and loading
- IORegistry information
    - GUID (64-bit Globally Unique ID)
    - Vendor ID, Model name, Configuration ROM
    - Unit Directories: Protocol information

AppleFWOHCI and AppleLynx

- Device drivers for PCI FireWire Link silicon
- AppleLynx: Blue and White G3; PCI-graphics G4
- AppleFWOHCI: Everything else
  - 1394 Open Host Controller Interface (1.0)
  - All add-in cards must use OHCI and fully comply with OHCI 1.0 and 1394a-2000
  - No opportunity to subclass or extend

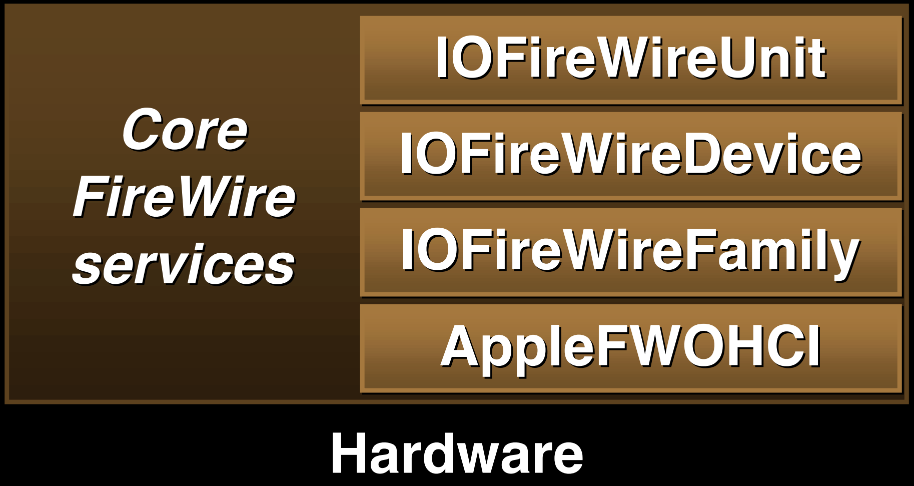
Additional Services

- FireWire User Client ("Device Interface")
  - IOFireWireLib.plugin
- SBP-2 Family and User Client
  - IOFireWireSBP2.kext
  - IOFireWireSBP2Lib.plugin
- AVC Family and User Client
  - IOFireWireAVC.kext
  - IOFireWireAVCLib.plugin

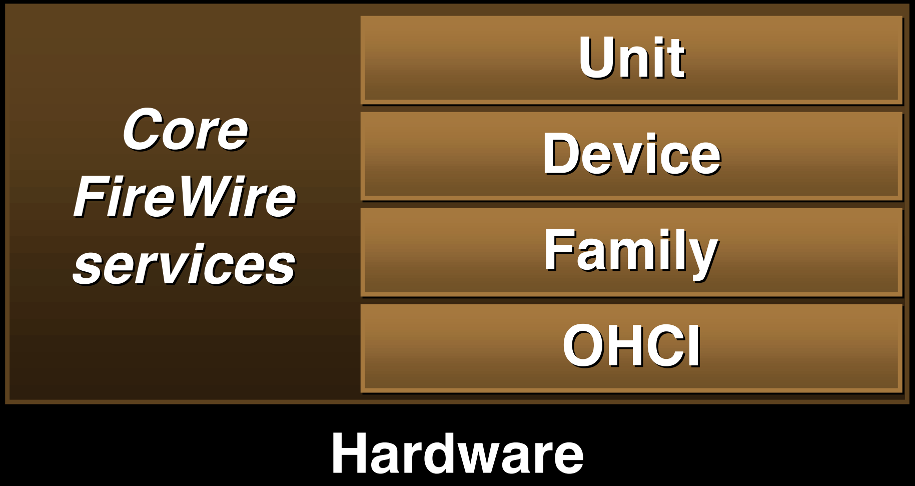
Everything Has a User Client

- All FireWire services are available in User space
- Writing drivers in User space is much easier
    - Crash one app, instead of kernel panic
    - User-level debugging tools
    - Reload/rerun without reboot
- Most developers should use User Client services instead of writing kernel FireWire drivers

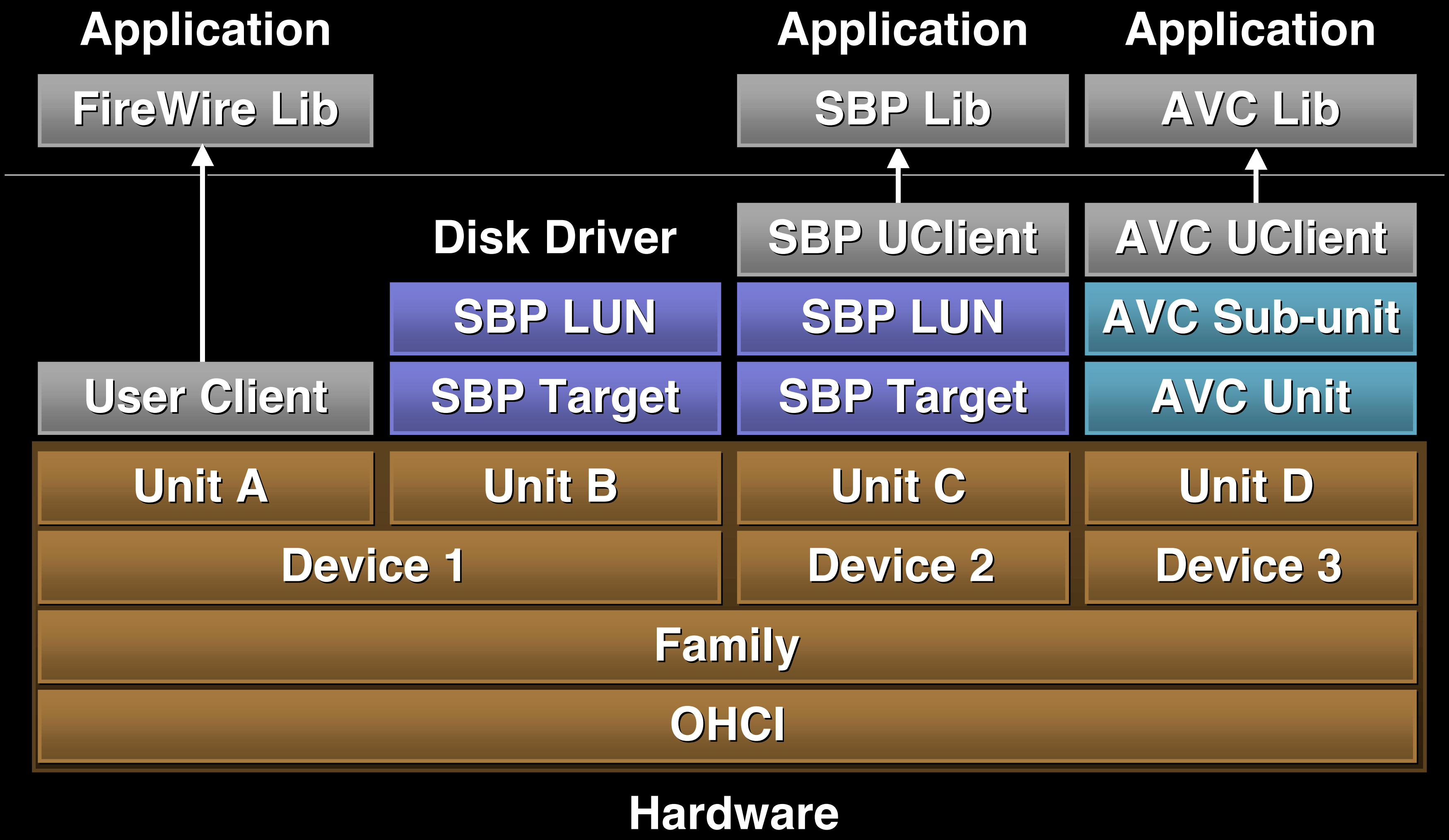
Additional FireWire Services



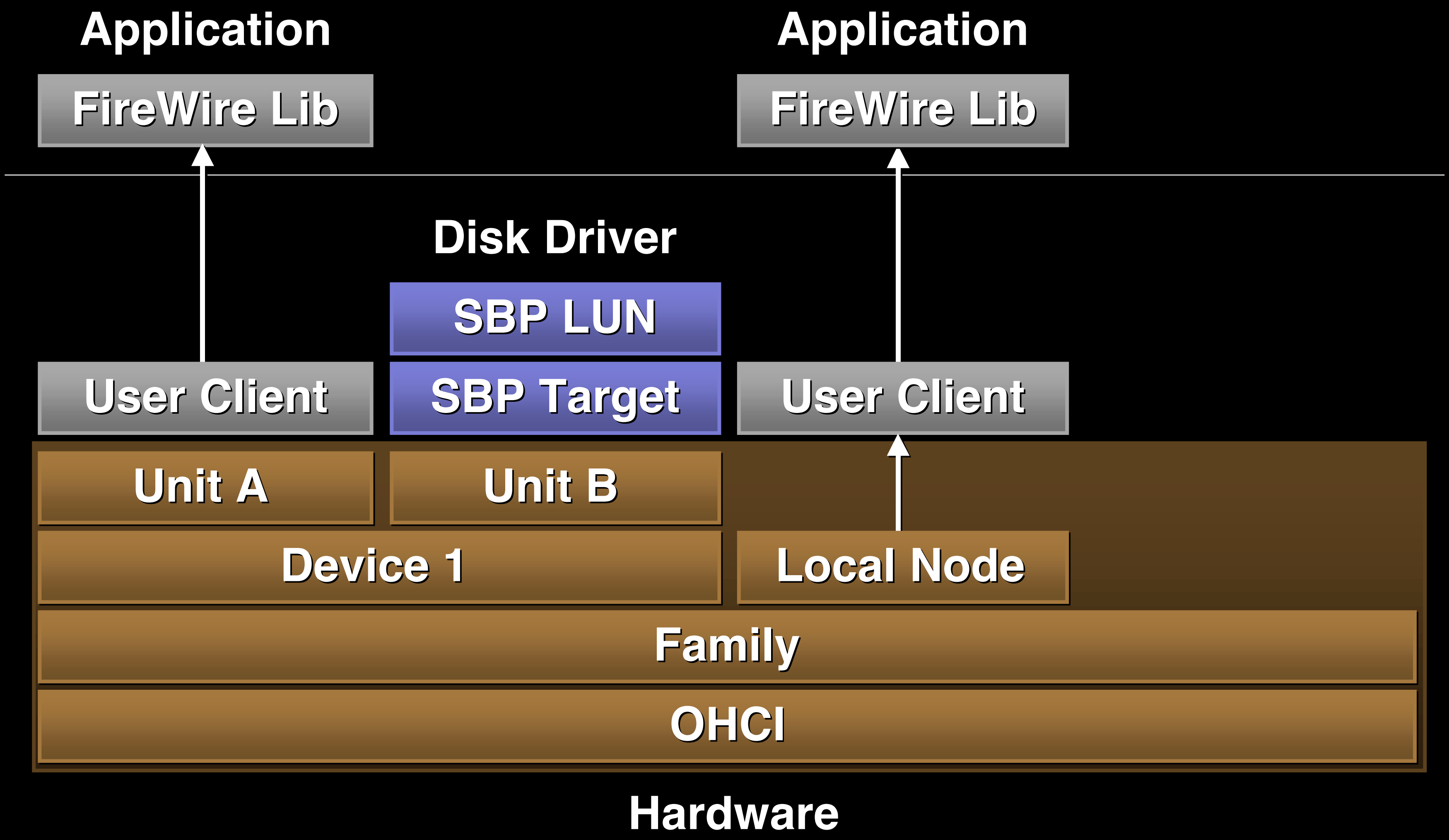
Additional FireWire Services



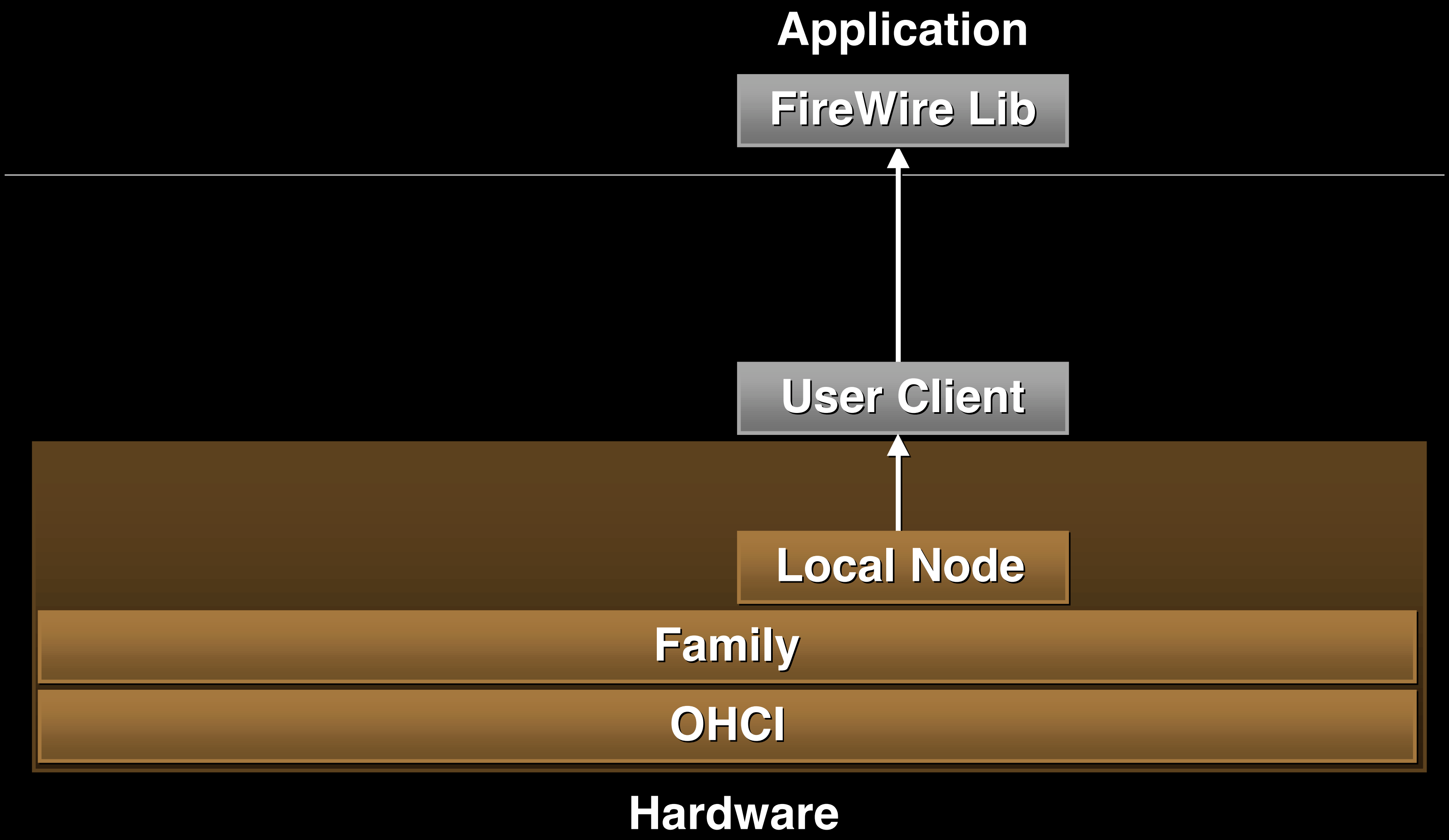
Additional FireWire Services



Additional FireWire Services



Additional FireWire Services



Application

FireWire Lib

User Client

Local Node

Family

OHCI

Hardware

FireWire
Changes Since WWDC 2001

New: SCSITask User Client

- Many devices using SBP-2 are SAM compliant (SCSI Architecture Model)
- SBP-2 User Client could access these devices
- SCSITask User Client is a better way
    - Much more high level
    - Hides low-level FireWire issues
    - Concentrate on your CDBs

New: Isochronous in User Client

- Core FireWire User Client (IOFireWireLib.plugin) now has full Isochronous services
- Send and receive isochronous streams with DCLs
    - Packet-by-packet abstract DMA program
- Direct memory map to DMA—no copies needed
- Sample code: IIDC (DCAM) camera driver

New: Better AVC User Client

- Full service—enables moving DV driver out of kernel (IOFireWireDV.kext)
- FCP (Function Control Protocol) in/out
- Plug Control Register service (PCR/CMP)
- Subunit enumeration in IORegistry
- Asynchronous Connections support

New: SBP-2 Features

- Retry on ack_data_error for PCI-Lynx
- New for Mac OS X only
    - Performance tuning flag: failsOnAckBusy
    - Optional Physical ORBs
    - Unlimited Page Table size
    - Login Retry with configurable delay

New: SDKs

- 5 new FireWire SDKs since WWDC 2001
- Latest binaries, sources, sample code, tools
- Now being integrated with Mac OS X Developer Tools CD
- SDKs are our biggest product for FireWire Developers
    - Feedback desired at Feedback Forum

New: HeaderDoc

- Almost 100 methods, classes, etc. documented since WWDC 2001, including:
    - IOFireWireDeviceInterface
    - IOFireWirePseudoAddressSpaceInterface
    - IOFireWireLocalUnitDirectoryInterface
    - IOFireWireLibPhysicalAddressSpaceInterface
    - IOFireWireCommandInterface

New: IIDC (DCAM) Driver

- Uncompressed, RGB or YUV video format
- QuickTime standard driver in Jaguar
  - High speed, large frame size
  - Much better than SDK sample code

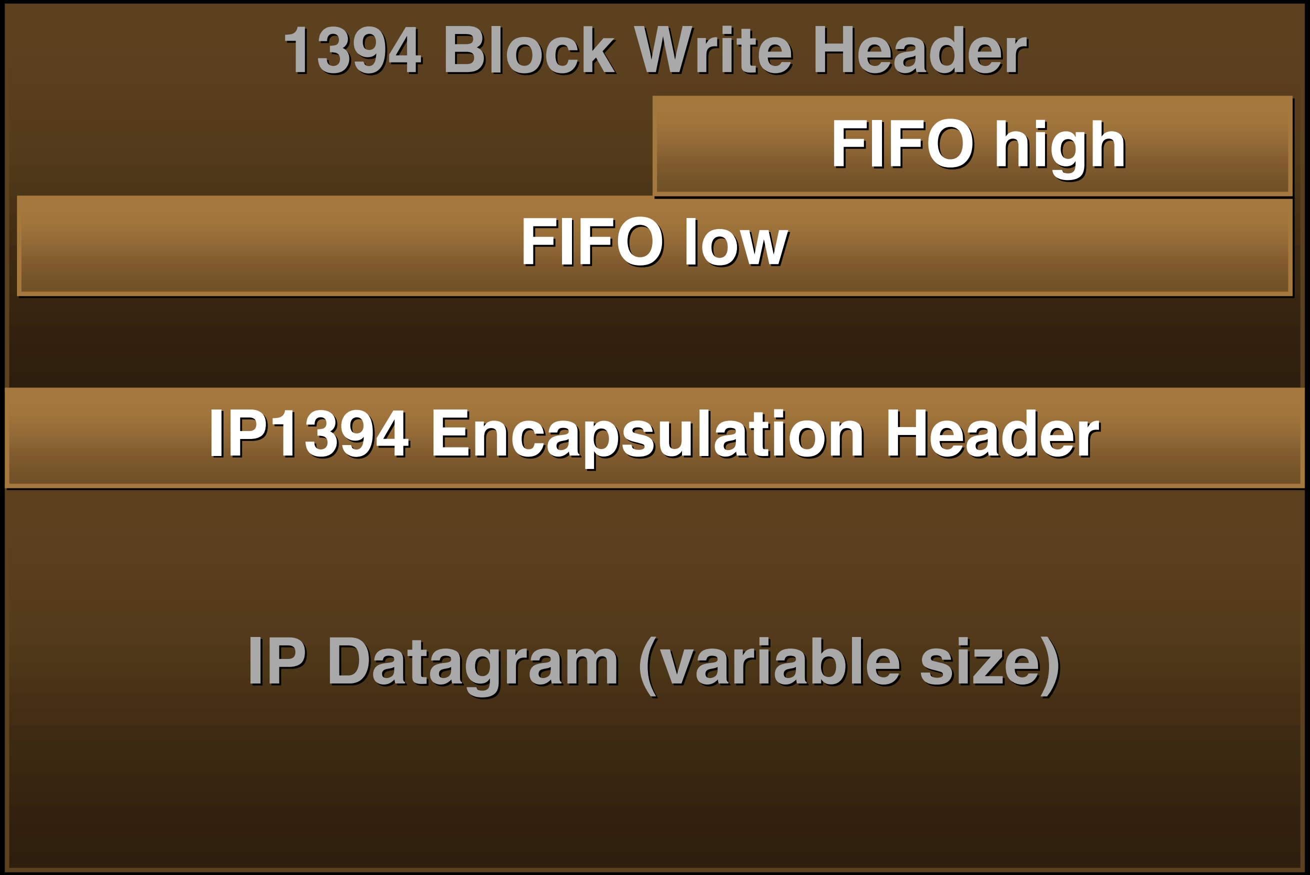
FireWire
Planned Future Services

Future: IP1394

- Internet Protocol driver for 1394
    - Per RFC 2734 (IPv4)
- Highlights
    - Send ARP packet to learn FIFO address
        - Sent in 1394a Asynchronous Stream packet
    - Send block write packets to FIFO
        - All packets go to the same address
        - Not a physical address!

IP1394 Datagram

- Encapsulated in 1394 Block Write packet



1394 Block Write Header

FIFO high

FIFO low

IP1394 Encapsulation Header

IP Datagram (variable size)

Future: Audio (ISO/IEC 61883)

- How fast is S400 FireWire?
  - Over 400 channels of CD-quality audio
- Mac OS X work
  - Audio Subunit and Music Subunit (AVC)
  - MIDI
  - SMPTE time code
- Contact Apple if interested

Future: IEEE 1394b

- IEEE 1394b changes are mostly at the PHY layer
- Most drivers and applications need no changes
- Minor low-level changes in Family and OHCI
    - New speed codes for DMA (800, 1600, 3200)
    - Larger asynch packets (4KB for all speeds)
    - More buffer memory
    - New topology information in Self-ID packets

IEEE 1394b Packet Sizes

Speed	Asynchronous	Isochronous
S100	512	1024
S200	1024	2048
S400	2048	4096
S800	4096	8192
S1600	4096	16384
S3200	4096	32768

Maximum packet sizes in bytes

IEEE 1394b Topology

10	phy_ID	0	L	gap_cnt	sps	br	c	pwr	p0	p1	p2	i	m

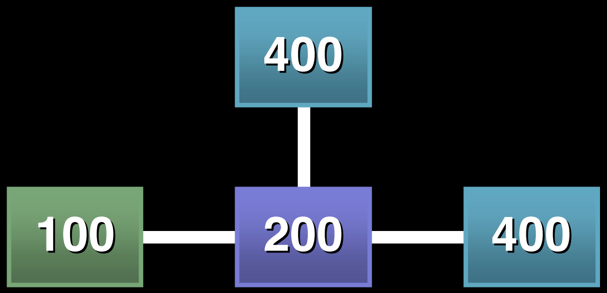
1394b self-ID packet (#0; first quadlet)

sp	PHY speed capability
00	S100
01	S200
10	S400
11	See port speed registers

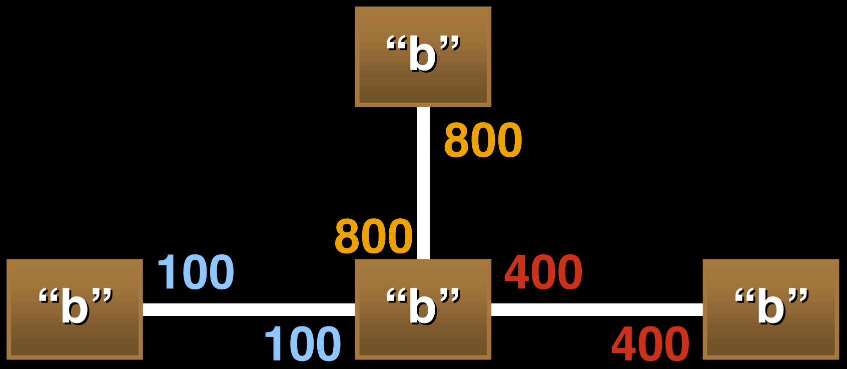
self-ID packet “sp” field values

IEEE 1394b Topology

- How to determine 1394b topology speeds?
  - Read PHY port registers to learn speed, or
  - Probe devices with asynchronous requests



1394a speeds: per-PHY



1394b speeds: per-port

FireWire
Device and Driver How-Tos

Device and Driver How-Tos

- What kind of device do I have?
- Do I need to write a driver?
- How do I match my device?
- How do I talk to my device?

How-To: Storage Devices

- Most storage devices supported by Apple’s driver
  - IOFireWireSerialBusProtocolTransport.kext
- Possibly subclass above kext, or other parts of the storage stack
- SCSITask User Client for most custom apps
- DiscRecording APIs (Session 008)
- Last resort: SBP-2 User Client, kernel driver

How-To: DV Devices

- Most DV devices supported by Apple's driver
  - Was IOFireWireDV.kext
  - Becomes IOFWDVComponents in Jaguar
- QuickTime Sequence Grabber
- AVC User Client
- Last resort: Subclass in kernel, or use FireWire User Client

How-To: IIDC (DCAM) Cameras

- Many devices supported by Apple's driver in Jaguar
- Or use Isochronous services in FireWire User Client
  - SDK has sample code, but it is inferior to Apple's driver in Jaguar

How-To: Scanners

- Consider Apple's Image Capture Architecture
- For SAM devices (device type = scanner) consider SCSITask User Client
- SBP-2 User Client
- FireWire User Client
- Scanner drivers do not belong in the kernel

How-To: Audio

- Apple is working on this
- Contact Apple to learn best approach

How-To: Printers

- If using SAM, consider SCSITask User Client
- If using SBP-2 (but not SAM), consider SBP-2 User Client
- If using IEEE 1394.3 (PPDT), consider SBP-2 User Client (and contact Apple)
- Consider FireWire User Client
- Printer drivers do not belong in the kernel

How-To: Still Cameras

- Some are/will be supported by Apple—contact us
- For SBP-2 devices, consider SCSITask User client (if SAM) or SBP-2 User Client (if not SAM)
- Not using SBP-2? You should be!
- For AVC devices, consider AVC User Client
- Or consider FireWire User Client
- Stay out of the kernel

How-To: PCI and CardBus

- Apple supports compliant Open HCI interfaces
    - AppleFWOHCI.kext
- Your OHCI silicon needs workarounds?
    - Get one that works
- Modifying OHCI silicon?
    - Contact Apple first

How-To: Hubs (Repeaters)

- Hub = Repeater
- Hubs do not need drivers in FireWire (unlike USB)
    - PHY silicon repeats packets automatically
    - There is no software configuration for hubs

How-To: Protocols

- Use the Local Node
- Access it from the FireWire User Client, or from the kernel
- Contact Apple

How-To: Other Devices

- Use the highest-level service possible
- If all else fails, use the FireWire User Client
    - Everything is possible, if not easy
- Stay out of the kernel if you can

FireWire
Developer Resources

Plugfest—Tonight!

- During Apple Campus Bash
- “Garage” room (above cafeteria)
- Test your FireWire devices
- Meet FireWire engineers
- Meet other FireWire developers

Plugfests—1394 Trade Assoc.

- Meet a much broader range of developers
- One-on-one testing for 2–3 days, plus “melee”
  - August 5–7 2002, Bellevue, Washington
  - October 2002, Taiwan
  - February 2003, USA West Coast?
  - April 2003, Tokyo?
- New 1394 TA Compliance Logo program
  - Test services from Quantum Parametrics

FireWire Kitchens

- A kitchen is 3–4 days with FireWire engineers and developers
    - Tutorials on latest FireWire services
    - Hands-on development and debug
    - Related presentations (e.g., PIMA/PTP)
- Cupertino and Tokyo, 2–3 times each year
- Often synchronized with a new SDK

Roadmap

101 FireWire Overview: Updates and future directions	Room A1 **Mon., 3:30pm**
111 Accessing SCSI and ATA Devices in Mac OS X: Including the SCSITask User Client	Civic **Wed., 5:00pm**
FireWire and USB Plugfest: Test your devices during the Apple Bash	“Garage” Room **Thurs., 7:00pm**
FF009 FireWire and USB: Tell us what to do next	Room J1 **Fri., 9:00am**

Who to Contact

Guillermo Ortiz
Apple Developer Relations, Technology Management
firewire@apple.com

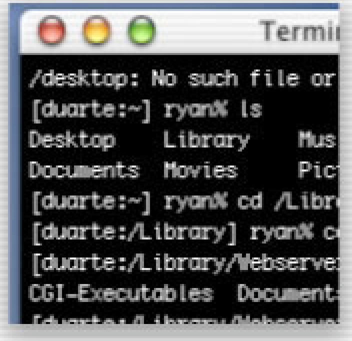
Public FireWire Developers mailing list
To subscribe, visit:
http://lists.apple.com

Public Mass Storage Developers mailing list
To subscribe, write "subscribe x_mass_storage" to:
requests@sam.apple.com

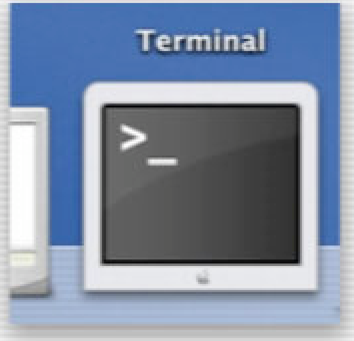
For More Information

- FireWire SDKs
  developer.apple.com/hardware/FireWire/index.html
- “Working With FireWire Device Interfaces”
  developer.apple.com/techpubs/macosx/Darwin/IOKit/iokit.html
- 1394 Trade Association
  www.1394ta.org
- IEEE
  standards.ieee.org

Q&A



```
/desktop: No such file or
[duarte:~] ryan% ls
Desktop     Library     Mus
Documents   Movies      Pic
[duarte:~] ryan% cd /Libr
[duarte:/Library] ryan% c
[duarte:/Library/Webserve
CGI-Executables  Document
```



Guillermo Ortiz
Technology Manager
firewire@apple.com

http://developer.apple.com/wwdc2002/urls.html

WWDC2002

WWDC2002

WWDC2002